Malware Analytics with YAKSHA

Constantinos Patsakis
University of Piraeus

21 November 2018 | InfoCom World

Problem setting

- Do all companies have security departments to monitor the development lifecycle of their products?
- How sure can you be that you know to which threats you are exposed to?
- Perform **independent** penetration tests on your platform!
  - Hire a company/consultant to do it.
  - Crowdsource it with bug bounty! Examples: Bugcrowd and HackerOne

Is this the best thing you can do?

- What about the truly malicious setting?
- The aforementioned ways will attract the “good guys”.
- A malicious adversary would keep the vulnerability **undisclosed** and **exploit** it **afterwards**.
- How do we monitor the “bad guys”?

YAKSHA

Honeypots

- A machine that looks vulnerable, but it is constantly monitored.
- An adversary who doesn't know that it is a honeypot, would attack it to take the "honey" out of it.
- They are used to gain **intelligence** about new techniques and exploits in the wild.

The goal of YAKSHA

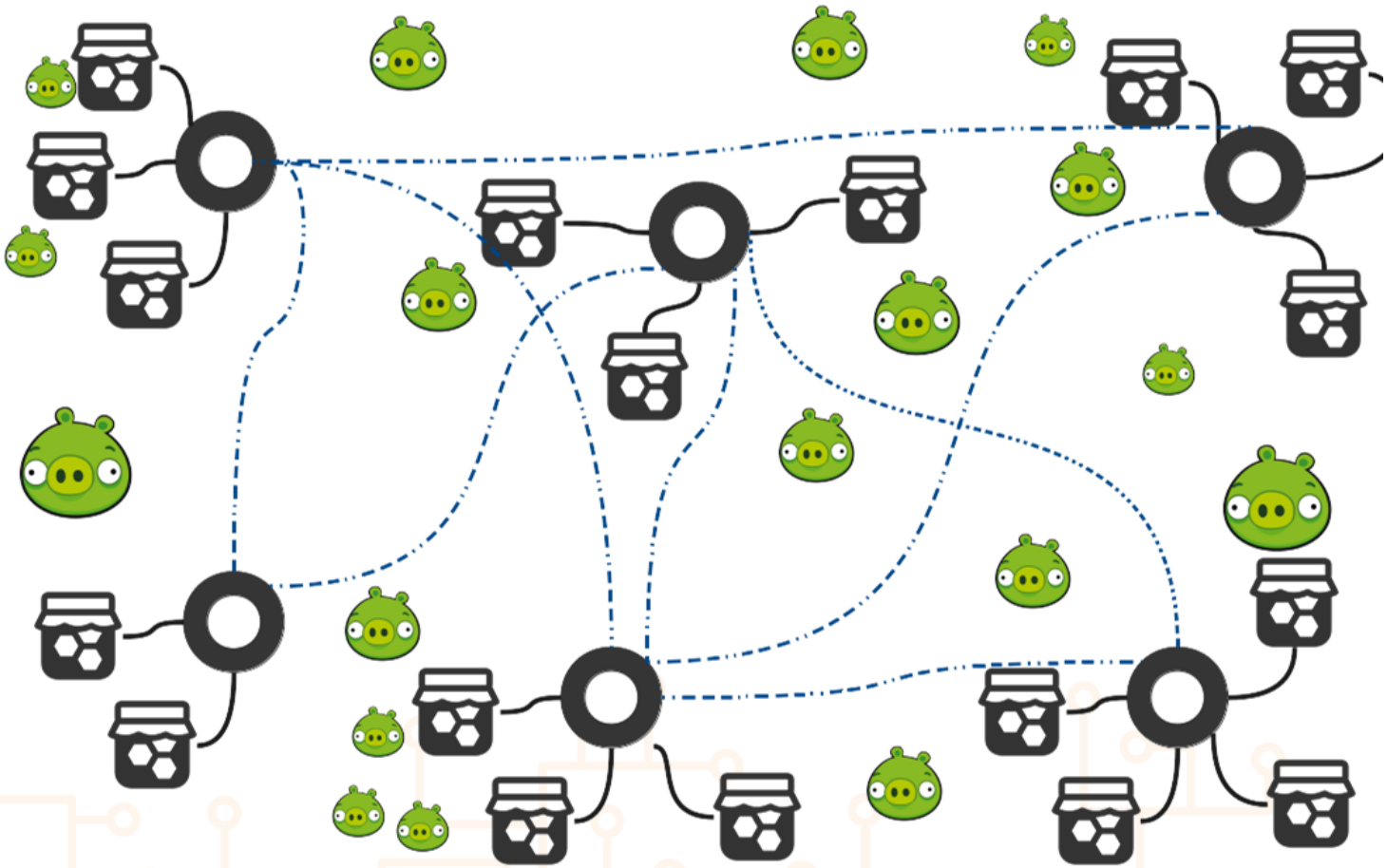
- Create a platform that will allow you to **automatically** create honeypots based on **your** settings.
- This way, you expose *clones* of your machines or your infrastructure, and monitor **what** a truly malicious adversary would do to it and **how**.
- YAKSHA will automatically create honeypots for:
  - Windows
  - Linux
  - Android
  - IoT devices

Output

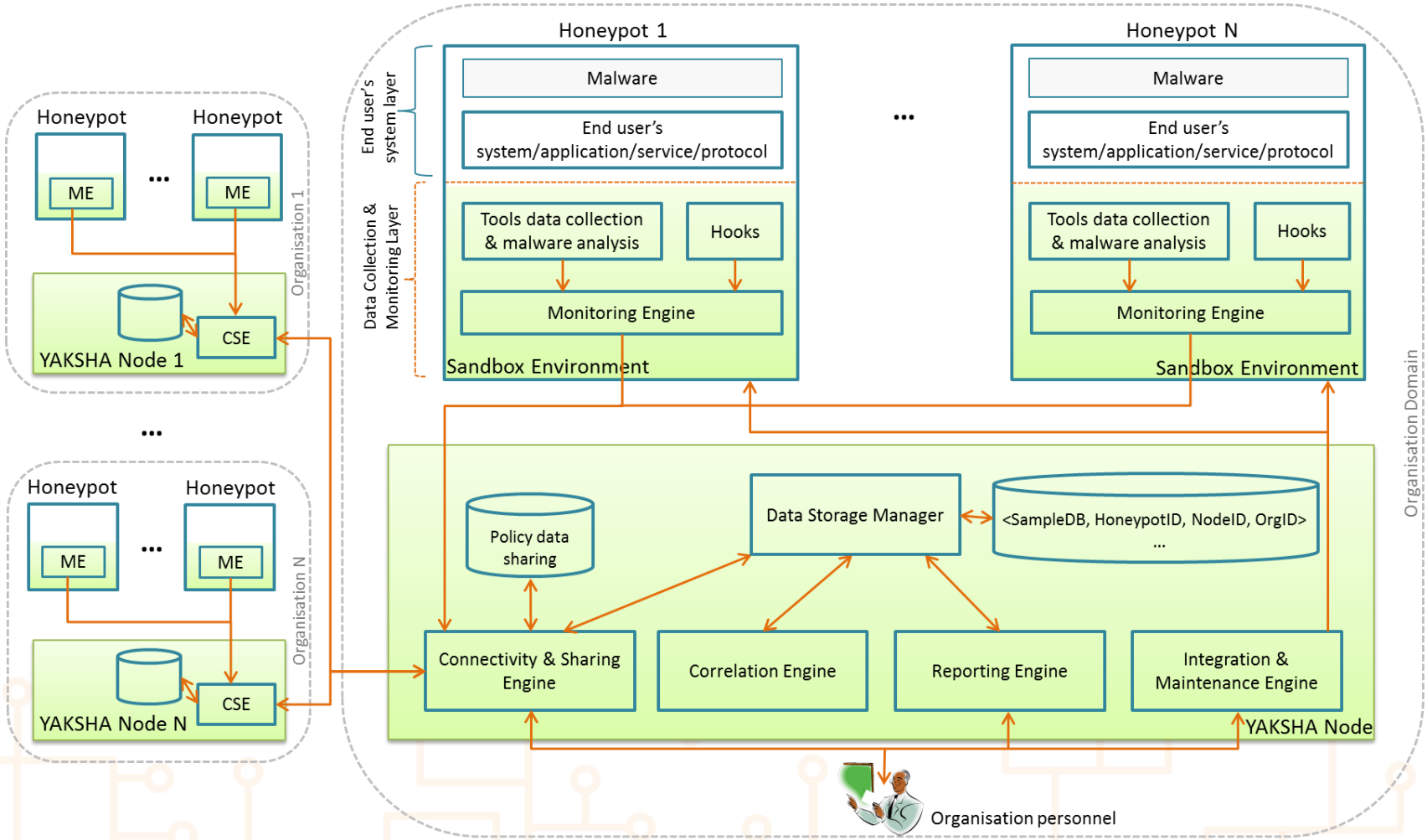
- YAKSHA will create reports for non-technical audience which will analyse to what risks they are exposed to.
- Full logs will be available for download and a list of proposed companies will be provided (monetization)
- Reports and logs will be provided based on user privacy policies
- Intelligence gained can also be shared (another monetization venue)

YAKSHA

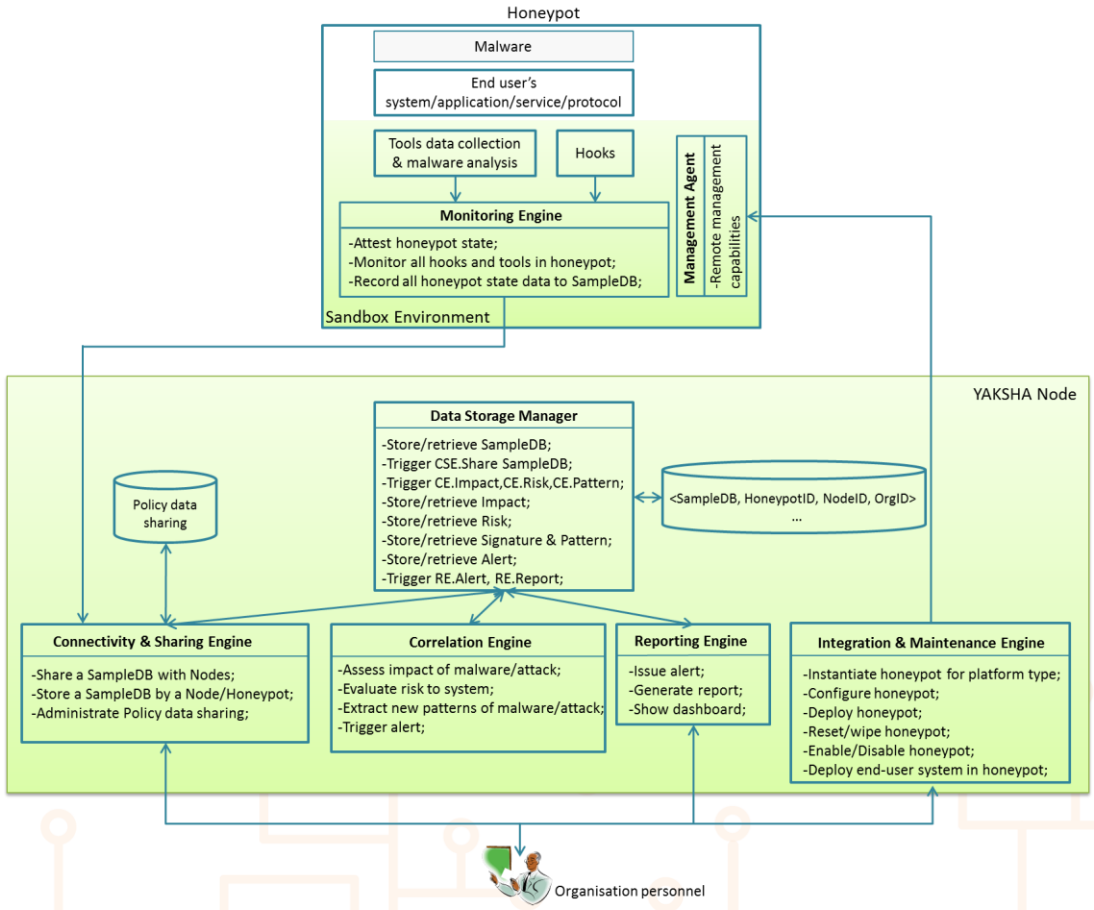
YAKSHA architecture



Architecture of a YAKSHA node



Architecture Functional View

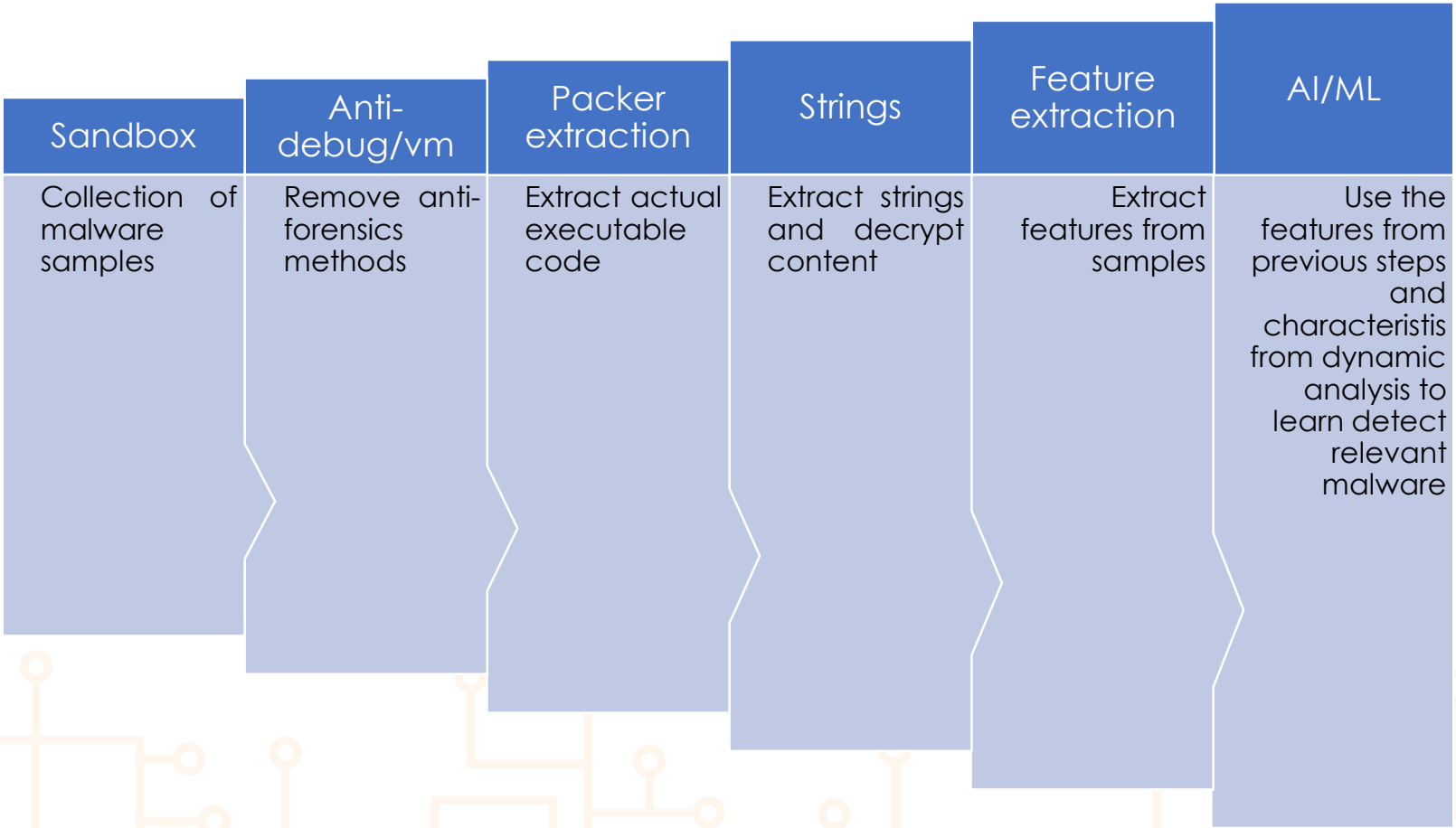


Technology/ tools

- Docker / Kubernetes
- Apache Mesos
- Jasper Reports
- BIRT Project
- ElasticSearch
- Cuckoo Sandbox
- DroidBox
- Qebek
- YARA
- Ansible
- Puppet
- Vagrant
- Honeysnap
- Sebek
- HFlow2
- MongoDB
- Conpot
- Glastopf / SNARE
- Kippo
- FLOSS
- FakeNet-NG
- packerid
- unxor , Xortool , XORBruteForcer
- BRO
- pev
- AnalysePE
- MASTIFF
- NetworkMiner
- ngrep
- tcpxtract
- Volatility
- TotalRecall
- Objdump
- Pyew
- Radare
- strace , ltrace
- Immunity Debugger
- Balbuzard
- Loki
- Malheur
- SeeTest
- angr
- Capstone
- yarGen
- Malfunction
- Libemu , scdbg
- Manalyze
- findaes
- python-evt
- python-registry
- Fabric
- Splunk
- Telnet IoT Honeypot

YAKSHA

Treating the collected malware



How does it work?

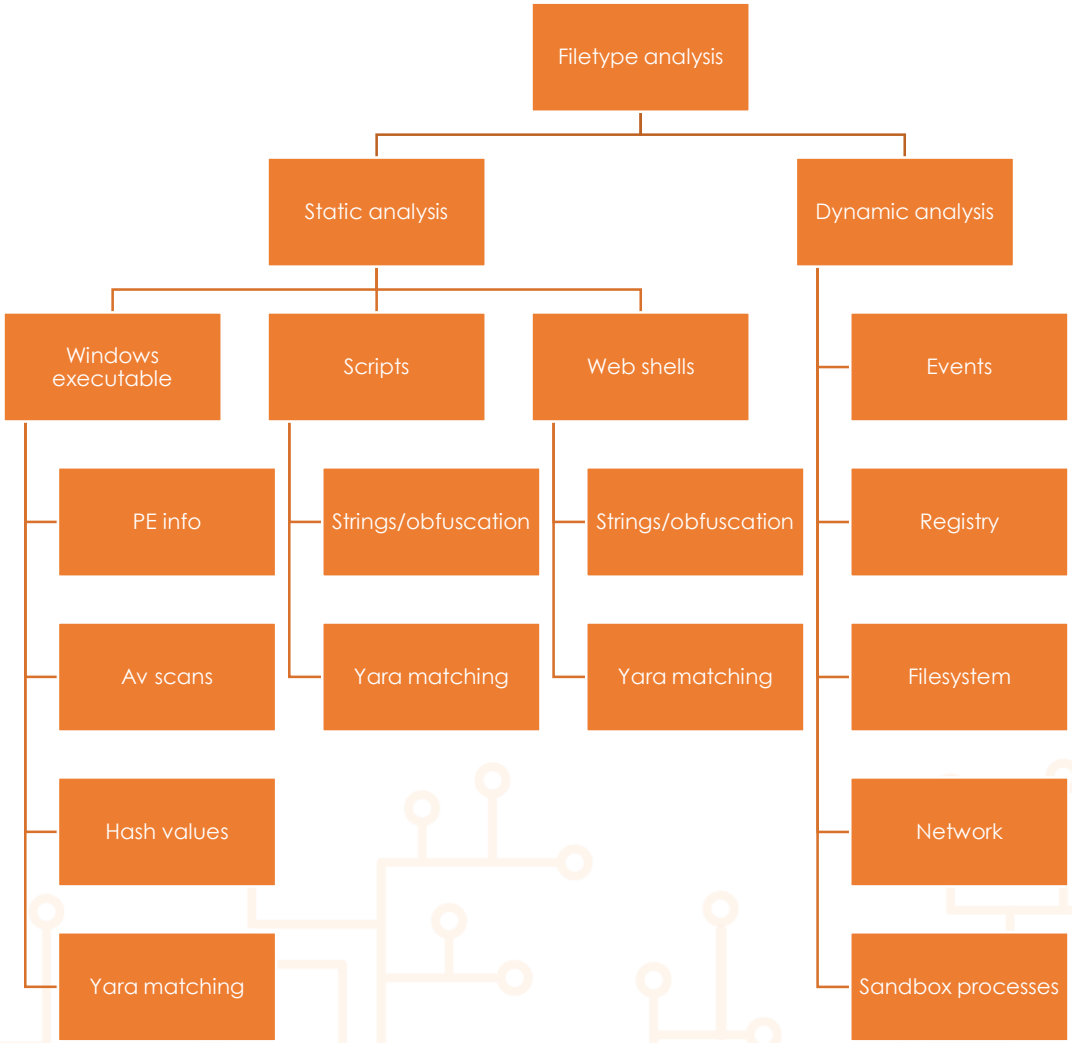
- We have created a UI that allows a registered user to create a VM (Windows/Linux/Android/SCADA)
- The user provides some initial configuration (RAM, HD space, cores etc) based on her purchased quota.
- YAKSHA provides the user with some login credentials to login the machine and configure it appropriately (install software, configure services etc.)
- When the system is ready, the user marks the system completed and the customized system is used as the prototype for honeypot deployment.

Collecting data

- The customized VM is equipped with monitoring tools to allow YAKSHA to record any attack that is performed to the system, capture all the commands performed, and collect every binary that is uploaded for further analysis.

YAKSHA

Malware analysis



Outputs

- Reports
  - Technical reports
  - Non-technical reports
- APIs
  - Share data/intelligence

YAKSHA

Thank you for your attention!

spi

UNIVERSITY OF PIRAEUS

OTE GROUP OF COMPANIES

inno

MOTIVIAN

VTT

Atos

vinasa

CyberSecurity MALAYSIA

NECTEC a member of NSTDA

ACIOA ASEAN Chief Information Officer Association

DISE

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 780498